Lab 4: Linear Lists

Required Materials

The following data files can be found as part of the standard Java library:

- java.util.ArrayList;
- java.util.Iterator;
- java.util.List;

Note: Links above are for OpenJDK and may not be identical to Oracle JDK.

Instructions

1. This lab can be completed individually or in a group (there is no size limit for the group), complete Exercise 1 and Exercise 2 below using the starting code project posted by your instructor. Your implemented data structures MUST work with the Driver.java included or you will receive a zero(0) on that component.
    - If working in a group, only one submission is required – **all group members' names must be included in the comments of the submission.**
        - All group members will receive the same grade.
        - It is your responsibility to manage the communication, cooperation and contribution of all group members.
2. See the *Marking Criteria* section below for details on how you will be assessed.
    - If there are more than one submission, both individually or across all group members, only the latest submission before the deadline will be accepted.
3. Submit your completed **zipped** exported Eclipse project to Brightspace by the posted due date.

Exercise 1

Write the Java code for the MyAList<E> class that uses an ArrayList<E> as its underlying data structure and includes the following methods:

- Constructor
- void add(E item)
- void addAll(MyAList<E> items)

- E get(int index)
- void remove(int index)
- void set(int index, E item)
- int size()
- boolean isEmpty()
- Object[] toArray()
- void clear()
- Iterator<E> iterator()

Exercise 2

Write the Java code for the MyDList<E> class that uses a separate class called MyNode<E> that stored a generic element and has a reference to its predecessor and successor. The MyDList<E> must contain a reference to the first and last nodes, as well as the size of the list. It should also include the following methods:

Note: Do not use any Java libraries for these methods, you must manipulate the links to the nodes and maintain the size of the list in your code!

- Constructor
- MyNode<E> getHead()
- void addFirst(E item)
- void addLast(E item)
- E removeFirst()
- E removeLast()
- size()
- isEmpty()

Marking Criteria

Criteria	Missing (0%)	Needs Improvement (0-50%)	Good (51-75%)	Excellent (76-100%)	Marks
Exercise 1	Not submitted	Significant MyAList methods are missing and/or not functional	Not all MyAList methods included and/or not functional	Most/all MyAList methods are included and functional	**/10**
Exercise 2	Not submitted	Significant MyDList methods are missing and/or not functional	Not all MyDList methods are included and/or not functional	Most/all MyDList methods are included and functional	**/10**
				Total	**/20**